Smart Sensors I

Multi-node Wireless Temperature Monitor

REFERENCE MANUAL

Group 164

Mike Schmitz
Joe Schneider
Rio Mascarenhas

ECE 403
December 16, 2002

Product Purpose

This project seeks to develop a wireless communication system for sensor detection, reporting, and communication. It entails periodic temperature collection from a wireless set of nodes for display and analysis on a portable user interface. The implemented communication system is an ad hoc network, which is a method of transmitting data from node to node in order to span large distances. The intended application for the finished product is food storage temperature monitoring. The device will collect temperature data from refrigerators and freezers in stores or on transportation vehicles. The device will then alert the user of any dangerous temperature levels to allow appropriate action to be taken to ensure food safety and quality are maintained.

Node Block Diagram and Specifications

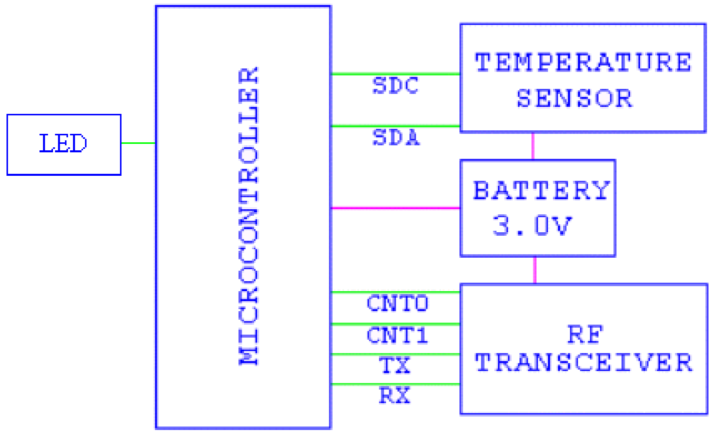

Figure 1: Node block diagram

The block diagram of a node is shown in Figure 1. The block on the left represents the microcontroller, a PIC16LF628, and associated discrete components to clock the microcontroller. The oscillator includes a crystal running at 4 MHz and two capacitors tied to ground. The microcontroller is able to communicate with the rest of the network by way of the RF transceiver, a RFM DR3000L. The transceiver is connected to the microprocessor through four lines: CNT0, CNT1, TX, and RX. The CNT0 and CNT1 lines are output only from the microcontroller, and are used to choose the mode of operation of the transceiver. Each CNT line is tied to V_{DD} with a 10kΩ pull-up resistor. Modes of operation for CNT0:CNT1 are given in Table 1.

CNT0	CNT1	Mode
Low	Low	Sleep (low power)
High	Low	OOK (On-Off Keyed) transmit
Low	High	ASK (Amplitude Shift Keyed) transmit
High	High	Receive

Table 1: Modes of operation for CNT0:CNT1 pins of DR3000 RF transceiver

The transceiver is used in OOK (On-Off Keyed) mode for transmissions, as this is more power-efficient than the only other transmission option on the transceiver, ASK (Amplitude Shift Keyed) mode. The TX line is an output only line from the microcontroller, and is tied to the transceiver through a 3.3kΩ resistor. The resistor is used to limit the current to the transceiver, which in turn controls the amount of power with which the transceiver transmits. The TX line must be kept at logic low when the transceiver is in receive mode. The RX line is an input only digital line to the microcontroller, representing the demodulated signal received by the transceiver. The only other discrete component used for the transceiver is a 33kΩ resistor which can be optionally connected to pin 8 of the transceiver (the LPF_ADJ pin) to allow for up to 19.2

kbps data bandwidth. Without a resistor attached to pin 8, the transceiver can only transfer data at up to 2400 bps. This resistor changes the speed of the RF link by adjusting the characteristics of the built-in low-pass filter of the transceiver. Further information can be found in the DR3000 data sheet.

In order to simplify the hardware and code complexity involved with communicating wirelessly, the RX line is tied directly to a hardware USART included on the PIC16LF628. This allows the microcontroller to run other code while the bytes of the packet are clocked in. The TX line is not connected to the hardware USART, however, since the hardware USART idles at a logic high. Letting the TX line idle at logic high causes the transmitter to continuously transmit, which wastes valuable battery power. Instead, a software serial transmitter written to emulate the hardware USART is implemented. The idle high required by the hardware receiver is padded to the beginning of all transmissions, fooling the receiving node's hardware receiver into thinking an idle high is being applied.

The temperature sensor, a DS1631, is connected to the microcontroller over a standard I2C two-wire bus. Both lines are tied to V_{DD} by a 10kΩ pull-up resistor. The SDA line is a bi-directional line over which all data is transferred. The SDC line is a clock line controlled by the I2C master, which is the microcontroller. All address pins A0-A2 are tied to ground, resulting in an address of 0x90 after shifting the address left one bit as required by the I2C protocol. The DS1631 can convert temperatures with precision from 9-12 bits, with more bits requiring more time to convert. After the microcontroller requests that the sensor begin converting a temperature, it can determine when the conversion is done by polling the sensor and reading a flag bit. When the flag changes to represent a complete conversion, the microcontroller requests the value of the newly converted temperature, which is then transmitted to the microcontroller.

The LED has been added to the node for debugging and demonstration purposes. It turns on for 250ms when the node is transmitting data. It operates on negative logic.

Base Unit Block Diagram and Specifications

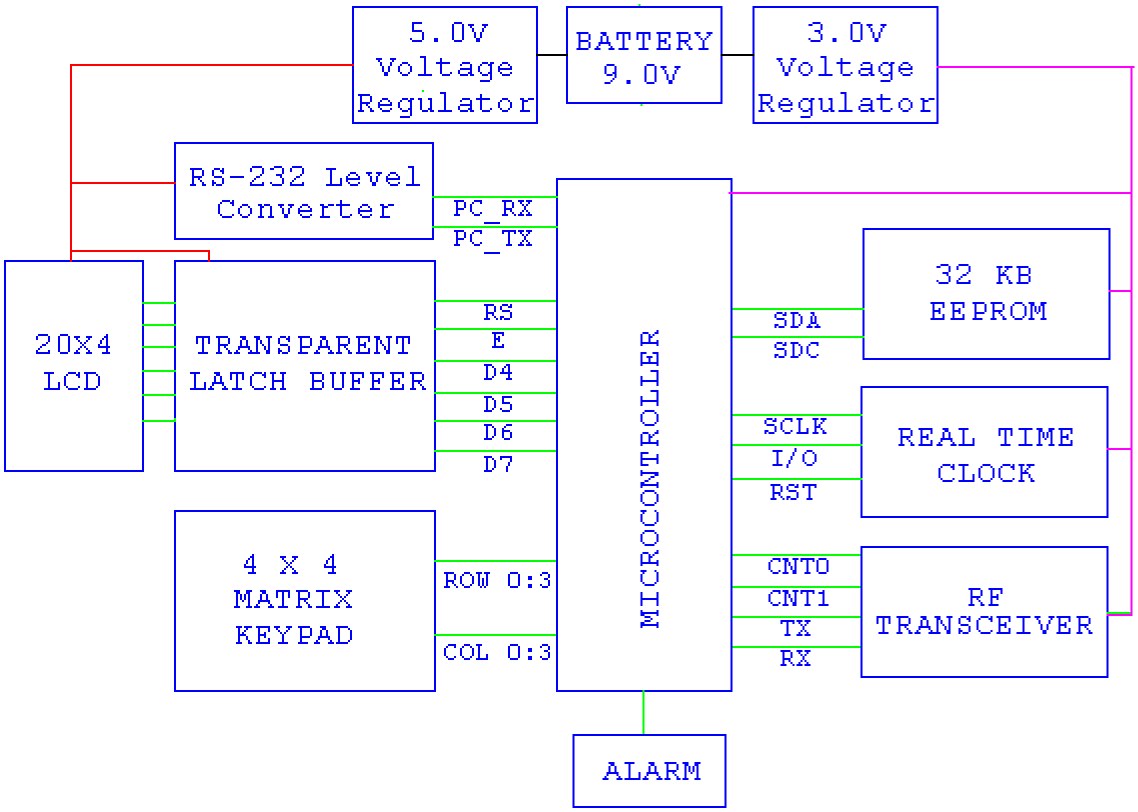

Figure 2: Base unit block diagram

The base unit block diagram is shown above in Figure 2. The sections of the base unit block diagram which overlap the node block diagram, namely the microcontroller oscillator components and RF transceiver, will be omitted here. The one difference that should be noted here from the nodes is that the base unit microcontroller is a PIC16F877. In addition to the components overlapped by the nodes, the base unit incorporates some components for data storage, accurate time keeping, and data input / output. Furthermore, all components on the base unit operate at 3-volts except for the LCD, transparent buffer latch, and RS-232 converter. The lower operating voltage helps to reduce power consumption.

In order to store data colleted from the network, the base unit includes a 32 kilobyte EEPROM, the 24LC256. This unit is connected to the microcontroller through the standard I2C two-wire bus. This is the same bus used to interface to the temperature sensors in the nodes, and thus both the SDA and SDC lines in the base unit require the same 10kΩ pull-up resistors that the nodes required. All address lines A0-A2 are tied to ground, giving the EEPROM an I2C address of 0xA0 after shifting the address left one bit as required by the I2C protocol. The SDA and SDC lines are tied to the built-in I2C hardware on the microcontroller. The microcontroller in the base unit, as in the node, acts as the I2C master, providing the serial clock on the output-only SDC pin and receiving and sending data over the bi-directional data line, SDA.

The real time clock is used to accurately keep the time at which temperature readings are made. These times are then stored along with the temperatures in EEPROM. The real time clock used, the DS1302, requires a 32.768 kHz crystal to keep the time accurately. The time is read and set via a two-wire serial bus. The SCLK line is driven by the microcontroller to provide the serial clock, and the I/O line provides a data path between the RTC (real time clock) and the microcontroller. The RST line is an active-low line which is left at logic low while not accessing the RTC to save battery power. The RST line must be driven to logic high in order to communicate with the RTC.

The matrix keypad, used for data entry in the base unit, is connected to the microcontroller through 8 lines, 4 of which represent the 4 rows of the keypad, and the other 4 of which represent the 4 columns of the keypad. The microcontroller reads the keypad by driving each column line high individually, while keeping all others at logic low. By reading the row lines as inputs when a particular column is driven high, the microcontroller can deduce which key is being pressed. The microcontroller scans through the columns whenever input is expected, waiting for a key press to be detected. Internal weak pull-up resistors are used in the microcontroller on the input lines in order to eliminate additional external resistors.

The 20 character x 4 line LCD provides a means of communicating retrieved data from the network to the user. It uses a standard Hitachi 44780 compatible driver IC, and is controlled in the same manner as other LCDs implementing a 44780 driver IC. As many LCD's require, contrast control must be connected to the module with the use of a variable resistor. The contrast pot is accessible inside the base unit case if adjustments are necessary. The LCD module used requires 5-volts to operate properly, and since the microcontroller, which operates at 3V, needs to drive the lines of the LCD module, a transparent latch buffer was placed between the LCD modules and the microprocessor. The transparent latch buffer, a SN74HCT573, translates the 3-volts logic high of the microcontroller to the 5-volts logic high the LCD module requires. The latch buffer is not required to operate in a transparent mode but does so since the active-low OE (Output Enable) is tied to ground and the active-high LE (latch-enable) is tied to high.

The RS-232 level converter is a standard MAX232N IC. This chip converts the idle-high TTL serial signals of the microprocessor to industry standard EIA-232 levels for communicating over a serial link with a PC. The MAX232N generates the negative and positive voltages required for EIA-232 by implementing a charge pump, which requires several capacitors. Furthermore, an PNP transistor is used on the PC_RX line between the MAX232N and the microcontroller to lower the voltage from 5 V to 3 V to comply with the microcontroller input specifications.

The purpose of the alarm is to alert the user of collected temperatures that fall outside of a user definable range. The buzzer that was chosen operates on DC voltage in the range of 3 to 12 volts. A 5-volt supply, rather than the available 9-volts, was chosen for the buzzer because the generated sound was considered efficient. Since the microcontroller could only provide 3-volts, a NPN transistor was used to switch in the higher voltage.

The power supply circuitry on the base unit consists of several capacitors, a 3-volt regulator, a 5-volt regulator, and a DPST switch. The regulators are necessary to produce the two required voltages in the base unit, while the capacitors are used to smooth out the produced voltages. A large 220uF capacitor is used across the input to the power supply circuitry to maintain a constant input voltage during battery/DC adapter source changes. A 0.1uF capacitor is also used to filter out any noise on the input voltage. The output of the 5-volt regulator has a 10uF capacitor to help prevent voltage drops during high current draws. Furthermore, the 3-volt regulator has a 100uF across its output for the same reason. A larger capacitance was required on the 3-volt side because circuit power ups caused the generated 3-volts to drop below 2-volts and thus caused the RTC to reset. The DPST switch is used to turn the circuit on and off. One pole controls the 5-volt supply to all the necessary components, and the other pole controls the 3-volt supply to the remaining components. The 3-volt power supply for the real time clock (ds1302) is not affected by the switch because it must always remain powered to keep accurate time.

Base Unit User Interface Overview

The user interface on the base unit consists of a LCD and keypad. The user operates the base unit by navigating through the menus on the LCD. The different options are explained below.

Node temp –
This option performs a single temperature collection from one node in the network. The user is prompted to enter the desired node's address, a temperature request will be made for the node, and the corresponding temperature will be displayed on the LCD. If communication fails with the node, "Communication Failure" will be displayed on the LCD. The collected temperature will not be stored in memory and is not available for download to a computer.

Start collection –
This option performs periodic temperature collection from a group of nodes. All collected temperatures along with the times of collection will be stored in memory and are available for download to a computer. If any temperature is outside of the temperature alarm range, the alarm will be activated. When this option is selected, the user will be prompted to enter the addresses of all nodes that should be polled, followed by the polling frequency. The polling frequency can range from continuous to once every 256 minutes with a resolution of one second. Once temperature collection begins, the LCD will display only the most recent collected temperature, the address of the node it came from, and the time of collection. However, if a collected temperature falls outside the temperature alarm range, only this temperature will be displayed on the LCD. It will remain on the LCD until replaced by a new temperature that falls outside the temperature alarm range.

View Data-
This option allows the user to view on the LCD the temperatures stored in memory during periodic collection. The user will be prompted to enter the address of the desired node, and then the stored temperatures and collection times for that node will be displayed on the LCD in the order of collection beginning with the most recent.

Connect –
This option allows a computer to control the base unit. The computer is able to download the temperatures and times stored in memory, download the routes to nodes in the network, read and set the base unit's clock, and perform single temperature collections from individual nodes.

Set clock –
This option allows the user to read and set the base unit's time and date.

Temp alarm range –
This option allows the user to read and set the minimum and maximum acceptable temperatures. If any collected temperature falls outside this range, the alarm will be activated.

Delete all memory –
This is the only way to delete the collected readings from the base unit's memory. This is accomplished by resetting the counters used to locate stored data for nodes.

Presets, Data Storage, and Clock Functions

The base unit presets, such as the minimum and maximum temperature alarm levels and the node buffer, which contains the nodes identified for polled, periodic collection, are set during the first power up of the microcontroller after programming. Any subsequent changes to these data values will be stored in EEPROM and will remain available even after power to the base unit is cycled. The default values, and the allowable ranges are shown in Table 2.

Value	Default	Range	Notes
Min. Temp.	0° C	-127° to 127° C with 1° divisions	If the selected min. temp is greater than or equal to the max. temp, the values will not be stored in EEPROM and the previous values will be retained.
Max. Temp.	100° C	-127° to 127° C with 1° divisions	
Node Buffer	Empty	Any node with an address less than the MAX_NODE can be added to or deleted from the node buffer.	The value of the MAX_NODE is hard coded in the microcontroller and cannot be changed. The default value is 4, thereby allowing up to 4 nodes to be polled.

Table 2: Preset values

The base unit will only store data that is collected during polled, periodic collection. Every temperature that is acquired from a node during this routine will be stored in EEPROM along with the time of reception and node of origin. The EEPROM dedicated to recording this information is 32 Kbytes in size, however some of this memory is dedicated to other tasks. To determine the number of readings per node that can be stored in memory, use Equation 1. The value of MAX_NODE is hard coded with a value of 4. It is beneficial to keep MAX_NODE as small as possible as this allows more readings per node to be saved. Once the memory is full, the oldest stored readings will be replaced with the new incoming readings. Therefore, the most recent measurements will always be available.

$$\text{\# readings per node} = \frac{32768 - 5 \times (MAX\_NODE) - (MAX\_NODE)^2}{8 \times (MAX\_NODE)} \qquad \textit{Equation 1}$$

The base unit has an internal real time clock to maintain accurate timekeeping of incoming temperatures. Once the clock is set, it will remain correct even after the base unit is switched off. To prevent the clock from resetting while changing the 9-volt battery, there are two options. The first is to plug the base unit into the DC power adapter to maintain power while removing the battery. The second, if the DC power adapter is unavailable, requires the base unit to be first turned off. The battery can then be successfully changed as long as the battery is not disconnected for longer than 7 seconds.

If the base unit loses all power entirely, all data and presets will be preserved automatically. Only the clock will be reset.

Wireless Communication Strategy

Communication in the wireless ad hoc network relies on two types of packets: Route Discovery and Route Aware. Route Discovery packets are responsible for finding the quickest route to any given node in the network, and Route Aware packets use the routes identified by Route Discoveries to transfer information in the network. The network operates on a poll-response strategy with the base unit initiating all communication.

When the base unit needs to acquire a temperature from a node, it searches its memory for a route to the node. If no route is available, the base unit transmits a Route Discovery packet as means to establish a route. The Route Discovery packet must propagate throughout the entire network in order to establish a route to any given node. This is made possible because any node that receives a Route Discovery will retransmit the packet as another Route Discovery. However, to prevent never-ending loops in the network, a node will only retransmit a given Route Discovery once. The ability to identify particular packets is accomplished with the Packet Code byte contained in every packet. As the Route Discovery diffuses through the network, it collects route information by having every node that retransmits it append its address to a route buffer in the packet. Therefore, every Route Discovery packet in the network contains a direct route back to the base unit. Once the desired node receives the Route Discovery packet from the base unit, it uses the enclosed route information to build a Route Aware packet to send back to the base unit. At the same time, it measures the temperature at the node and adds this information to the packet. The Route Aware packet will then follow the route identified in the route buffer to get back to the base unit. When the base unit receives the Route Aware packet in response to the Route Discovery packet is sent earlier, it will store the route information contained in the packet's route buffer in EEPROM memory for later usage. Likewise, the temperature at the node will be processed, displayed, and stored in memory.

If the base unit must acquire a temperature from a given node, and a route to that node is stored in memory, a Route Discovery packet is not necessary. Instead, the base unit will transmit a Route Aware packet with a route buffer containing the route to the node as means to request the temperature. Because the Route Aware packet does not need to propagate through the entire network, it consumes considerably less power from the network on a whole. Likewise, because of small transmitting delays designed to prevent cross-talk in Route Discovery communication, a Route Aware packet will reach the desired node in less time.

The algorithm used by the base unit to acquire a temperature from a node is shown by the flowchart in Figure 3. As shown by the flowchart, even though a route may be stored in memory for a given node, it may no longer work due to changing network conditions. Therefore, a Route Aware packet will only be used twice before the route is considered inoperable and a Route Discovery packet is used to establish a new route. Repetitive attempts at communication are necessary due to noise and cross-talk in the network which may cause some packets to be dropped and thereby terminate communication.

To prevent packet collisions and cross-talk in the network during Route Discoveries, nodes wait a random amount of time before retransmitting a packet during propagation stages. Therefore, when several adjacent nodes receive a Route Discovery packet at the same time, they won't all retransmit the packet at the same instance, which would cause interference and loss of data.

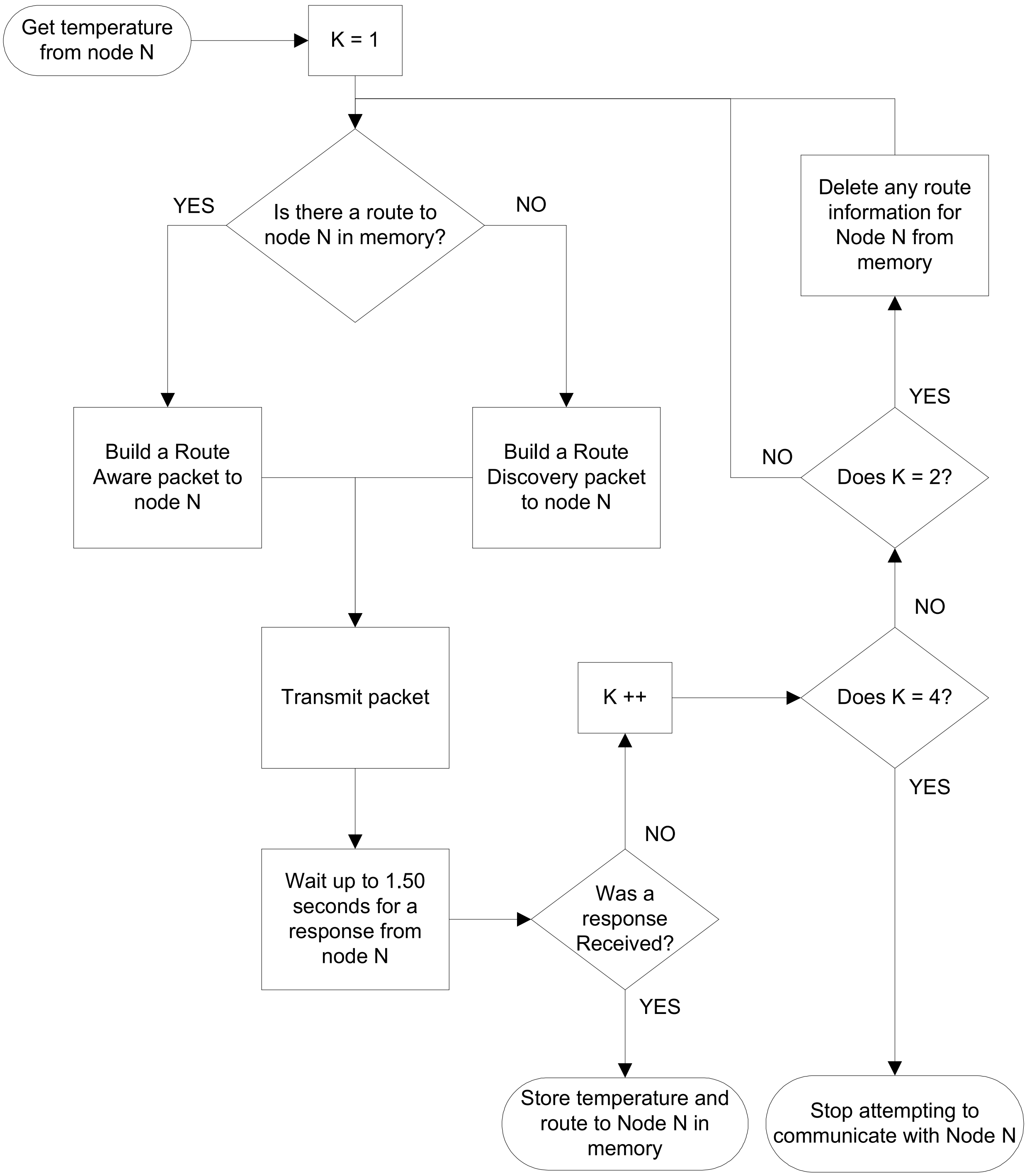

Figure 3: Wireless Communication strategy

The general format of the wireless packets is shown in Figure 4. Both the Route Discovery and Route Aware packets, except for a few minor changes, use this same format. The descriptions of each section of the packet follow.

Preamble	Address	Packet Code	Checksum	Length	Route Bytes	Data Bytes

Figure 4: General protocol format

Packet Section	Description
Preamble	The preamble is necessary for transmitter/receiver USART synchronization and receiver initialization. The first part of the preamble must consist of no less than 10 high bits, without any start or stop bits accustomed to USART communication. This causes the receiver's USART to reset and therefore perceive the next high to low transition as a start byte. The rest of the preamble consists of 3 bytes with the binary value "01010101." These three bytes are sent with the normal start and stop bytes and are necessary to charge the capacitor in the data slicer of the receiver to a specific level for optimum noise rejection during packet transmission. The three identical and consecutive bytes are also used to key the receiving node on the beginning of a packet transmission.
Address Byte	The address byte identifies the destination of the packet and the type of packet being transmitted. The most significant bit of the address byte is set if the packet is for Route Discovery, and it is cleared if the packet is for Route Aware. The lower seven bits are used to identify the address of the destination node. Thus, the protocol can accommodate up to 127 nodes and one base unit. If the packet is of type Route Discovery, the address contained in the address byte is the address of ultimate destination for the Route Discovery packet. On the other hand, the address byte identifies the next node in the included route if the packet is of type Route Aware.
Packet Code Byte	The packet code byte is necessary to prevent the flooding of the network with Route Discovery packets. By definition, a Route Discovery requires every node receiving a Route Discovery packet to update the packet's route cache with its own address and then retransmit the packet. However, this could create infinite communication loops which would clog the radio frequency and drain power. The packet code prevents this by including a random number in the packet to serve as a packet identifier. Therefore, if a node receives a Route Discovery packet but the packet code is identical to the previous packet code it received, the node will ignore the Route Discovery and not retransmit the packet.
Checksum Byte	The checksum byte is used for error detection in the packet. The checksum equals the exclusive- or of all bytes in the packet excluding the preamble and the checksum byte itself. The value of the byte is calculated by the transmitter, sent with the packet, recalculated at the receiver, and compared. If the checksums do not match, an error occurred in communication, and the receiver will reject the entire packet.
Length Byte	The length byte is necessary for the receiver to identify the number of routing addresses and data bytes contained in the packet, and the total length of the packet. The highest four bits of the length byte indicate the number of data bytes in the packet, anywhere from 0 to 15. Likewise, the lowest four bits of the length byte indicate the number of addresses contain in the route cache of the packet, anywhere from 0 to 15. With this information, the total length of the packet can be calculated as the sum of the address bytes in the route cache, the data bytes, the 4 overhead bytes, and the three bytes in the preamble.
Route Bytes	The route bytes comprise the packet's route buffer. During a Route Discovery, the route buffer is where a relaying node would add its address, and thus build a route structure for the receiving node. For a Route Aware, the route cache contains the addresses of the nodes the packet must follow to reach its destination.
Data Bytes	The data bytes are the reason for the packet in the first place, to transfer information. Any type of information can be stored in this location, up to a maximum of 15 bytes.

Base Unit/Computer Protocol

The PC software that is associated with the base unit requires a standard method of data transfer in order to successfully communicate with the base unit. A custom protocol has been developed for this purpose. It dictates how a connection is initiated, what type of functions can be performed, and the format for all transfers. Before any function can be performed, the PC and base unit must first establish a connection. This is initiated by setting the base unit to connect mode. The remaining connection sequence is shown below.

Connection sequence:

Purpose: To identify the firmware version on the base unit and ensure the PC software supports the firmware version.

Data Direction	Packet	Details
BU -> PC	'sIDV.vX'	V = major version number v = minor version number X = checksum
PC -> BU	'g' or 'b'	g = version supported b = version not supported

Once a connection has been made, the PC software can perform four functions. They are listed and discussed in detail below.

1. Real Time Clock functions
2. Temperature request from single node
3. Sensor route retrieval
4. Previous stored data retrieval

Real Time Clock functions:

Purpose: To set or receive the current time on the real time clock chip on the base unit.

Data Direction	Packet	Details
PC -> BU	'TR'	Read time from RTC chip on BU
BU -> PC	'sMDYHNSCX'	M = month D = day Y = year H = hour N = minute S = second *all variables for this packet are in BCD form X = checksum – not in BCD form

Data Direction	Packet	Details
PC -> BU	'TS'	Set time of RTC chip on BU
BU -> PC	'g'	BU ready to receive time
PC -> BU	'MDYHNSX'	M = month D = day Y = year H = hour N = minute S = second *all variables for this packet are in BCD form X = checksum – not in BCD form
BU -> PC	'g' or 'b'	Acknowledge the set time packet and assert that time has been set. Error with checksum, time will not be set

Sensor Data Packet:

Purpose: To request the temperature from a node. This packet causes the base unit to send a packet out to the sensor and wait for a response from the node.

Data Direction	Packet	Details
PC -> BU	'QN'	N = node to which packet is sent
BU -> PC	'sTtX' or 'b'	T = upper 8 bits of temp t = lower 8 bits of temp X = checksum Sensor did not respond

Sensor Route Packet:

Purpose: To acquire the route being used to communicate with a particular node in the network.

Data Direction	Packet	Details
PC -> BU	'RN'	N = node for which route is to be sent
BU -> PC	'sC(RD)X'	C = number of bytes in route data (RD) = route data X = checksum

Previous Stored Data Retrieval:

Purpose: To retrieve data stored in the EEPROM on the base unit, which would normally have been collected by the base unit before being connected to the computer. When connected to the computer, the computer stores all retrieved temperatures locally, not in EEPROM.

Data Direction	**Packet**	**Details**
PC -> BU	'DN'	N = node for which data is to be retrieved
BU -> PC	'sTtMDYHNSX' or 'e'	T = upper 8 bits of temp t = lower 8 bits of temp M = month D = day Y = year H = hour N = minute S = second X = checksum *all time variables for this packet are in BCD form *This must be ACK'd by the PC, see below. No more records available. After this is transmitted, communication between the PC and BU is finished. (The PC does not have to ACK this in any way.)
PC -> BU	'g' or 'b' or 'e'	Checksum of last record was correct Checksum of last record was incorrect, please retransmit. If the BU receives this response, it should retransmit the last packet until it receives a 'g'. Tells base unit to stop sending data.

Operating and Electrical Characteristics

Wireless Range:	
Ideal conditions	~50 feet
Obstructions / noise	~10 feet
Wireless Frequency:	916.5 MHz
Wireless Modulation:	OOK (On Off Keyed)
Operating range:	-40 °C to +70 °C
Temperature accuracy:	accurate to ± 1 °C
Max number of nodes:	127
Max number of hops:	16
Base unit memory:	32k
Power supply:	
Base unit	9V battery or 9V DC input
Node	2 x AA batteries
Battery Life:	
Base unit	~72 hours
Node	~360 hours, depending on frequency of usage
Physical dimensions:	(WxLxH)
Base unit	5 ½” x 7” x 2 ½”
Node	2 ¼” x 3” x 1 ½”
C Compiler	CCS COMPILER 3.060